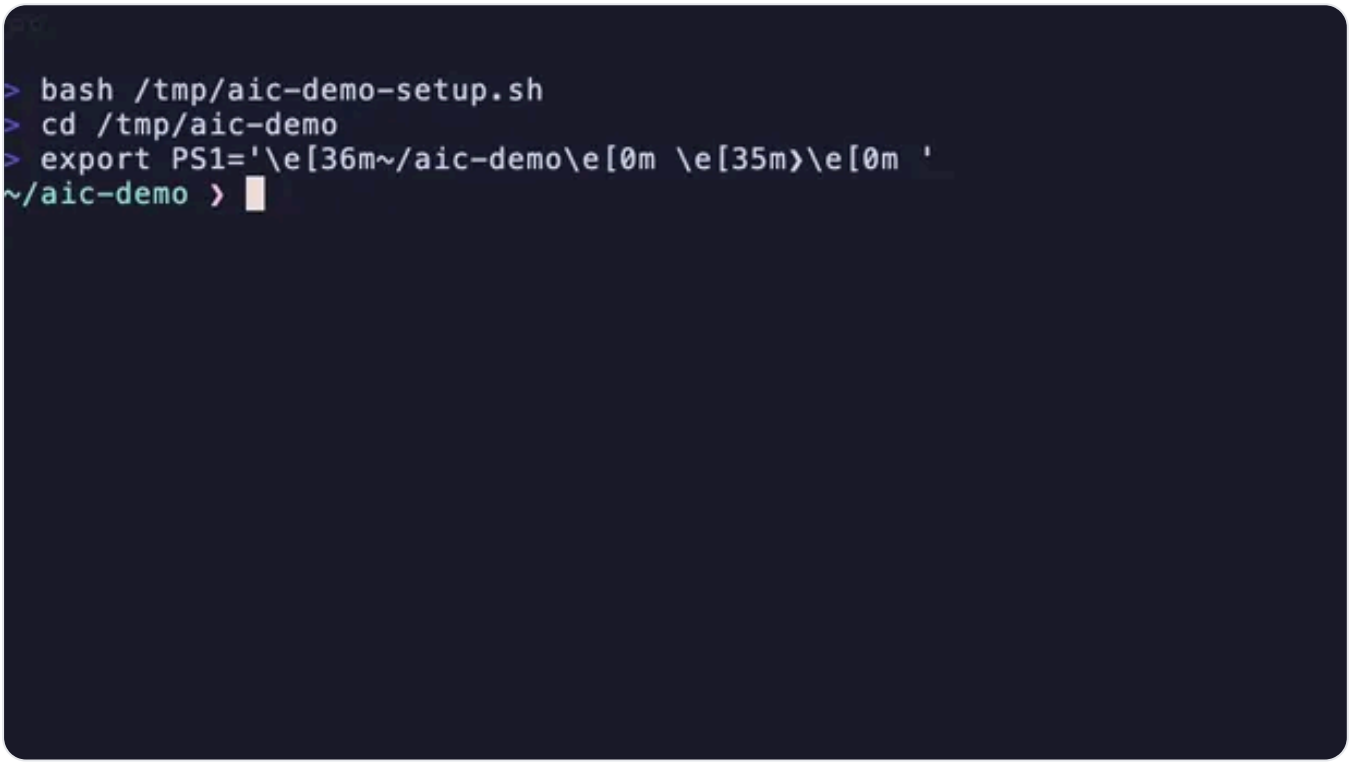
A

aic

AI-powered git commits that read your diff, draft conventional messages, and commit in one command.



About

aic is an AI-powered git commit tool that reads your diff, drafts a conventional commit, and commits it in one command. It also helps when nothing is staged by grouping work into logical commits, and it can walk through merge conflicts file by file with a review step before anything lands.

What is aic

aic is an AI-powered git commit tool built to turn a diff into a conventional commit with very little manual work. The app reads staged changes, drafts a message, and can create the commit from the command line in a single flow. When there is no staged work,

the tool can still help by batching unstaged changes into logical commits instead of forcing you to sort everything by hand.

The project is positioned as a cleaner way to ship commit history for developers who want faster writes without giving up readable messages. It supports a one-time setup step, then runs from the terminal as a local command. The site also shows that aic can handle mid-merge situations by resolving conflicts file by file and letting you approve a diff before changes are committed.

Key features

aic focuses on AI commit generation, commit batching, and provider-controlled local execution.

- Reads your diff and drafts a conventional commit message from the change set.
- Commits staged work in one command, which reduces the back-and-forth of manual message writing.
- Groups unstaged work into logical commits when nothing is staged, so you can clean up history faster.
- Supports mid-merge conflict handling with file-by-file resolution and an approval step.
- Uses your own provider and model rather than routing through a middleman service.
- Shows a resolved config with a masked API key through the `aic list` command.
- Supports providers named on the site, including OpenAI, Anthropic, Gemini, DeepSeek, Groq, Ollama, xAI, Mistral, OpenRouter, Perplexity, Together, and OpenAI-compatible setups.
- Installs through Homebrew on macOS and Linux, or via installer scripts for macOS/Linux and Windows.

The app is especially centered on AI-powered git commits, but it also covers the practical parts around setup, installation, and provider selection. That makes it useful if you want the commit helper to fit into an existing CLI workflow instead of replacing it.

How to use it

binaries.

Run `aic setup` once to choose a provider, enter your key, and pick a model.

Use `aic` on staged work to read the diff, draft the commit, and commit the result.

If you have no staged files, let the tool batch-plan the unstaged changes into logical commits.

When you need to inspect settings, run `aic list` to view the resolved config with a masked key.

For Homebrew users, update with `brew upgrade aic`; the site says `aic update` redirects to Homebrew when that install method is detected.

The workflow is intentionally short: install, set up, run, review, and ship. That makes AI-powered git commits feel closer to a native terminal command than a separate web app.

Who it is for

aic is for developers who want AI-powered git commits without changing their editor or hosting their source data in a separate service.

Solo developers who want faster commit writing from the terminal.

Teams that care about conventional commits and readable history.

People who often end up with unstaged work and want the tool to group it into commit-sized pieces.

Engineers who work through merges and want conflict handling with review before commit.

Users who already have an API key and model preference for an LLM provider.

Because aic uses your own provider, it fits users who already have access to OpenAI-compatible or vendor-specific models and want to point the tool at that setup. It is less about project management and more about making git history easier to produce.

What to know before you use it

aic is useful, but it is not a replacement for careful review. The app drafts conventional commit messages from diffs, yet the quality of the result still depends on how clear the change set is and which model you choose. If your diff is noisy, the suggestion may

A few practical caveats matter:

- It is a command-line tool, not a browser dashboard.
- You need to supply your own provider, key, and model during setup.
- The site says your API key never leaves your machine, but the tool still sends requests to the selected provider.
- It helps with merge conflicts by working file by file, but it does not remove the need to understand the conflict before approving it.
- If you need an authoritative history for compliance or release notes, review the final commit before relying on the generated message.
- For provider billing, model behavior, or key management, use the official provider console instead of the app.

In short, aic is a focused AI-powered git commit helper for cleaner history and faster terminal workflows. If your goal is to stop hand-writing every commit message while still keeping control over the final result, this tool is built for that exact job.

FAQ

What is aic?

How does aic generate commit messages?

Is aic free to use?

What AI providers can I use?

Which platforms does aic support?

Which AI providers are supported?

How does aic handle unstaged changes?

How do I install aic?

Does aic send my key through a proxy?

How does auto-batching work?

Does aic handle merge conflicts?

What does `aic list` do?

Sponsored

Belay – Awa... Featured
Developer Tools

Stays awake while your agents work.

Free # Developer Tools

View ↗

Partly Featured
Finance

Partly is a mobile app that lets friends quickly scan receipts, split any bill fairl...

Free # Finance

View ↗

Visit website ↗

aic

Pricing Free

Category AI

A aic

Launched	Aug 2026
Updated	18 days ago
Verified	1 months ago

TAGS

ai # git # commit # commits # automation # llm # cli
conventional-commits # developer-tools # open source # merge-conflicts

© 2026 Dir Hub

A aic